

Taller Express y otras cosas



Nombre: Octavio Sebastián Hernández

Matrícula: A01638993

Película favorita y razón: Interestelar por su historia, actuaciones, efectos visuales y banda sonora, además de su interpretación del espacio y de las cosas cuyo funcionamiento no ha podido ser comprobado por la ciencia.

El objetivo de la actividad es explorar y conocer más sobre el desarrollo web del lado del backend.

Al final de la actividad deberás hacer un PDF con este documento. El PDF deberá tener un enlace en supertarea.html

- **Ejercicio 0: El pasado**

En las semanas previas vimos la forma de consumir un API desde el cliente (el navegador). Ahora consumirás el **API desde el servidor**. Usa JavaScript y NodeJS.

Selecciona un API que no requiera autenticación.

El servidor deberá consumir el API, procesarlo de alguna manera (ejemplo: seleccionar un dato) y posteriormente entregar ese dato (o datos) al cliente (navegador).

Sube tu archivo a tu repositorio.

Escribe aquí abajo el enlace al archivo en tu repositorio:

<https://seb-hdez-g.github.io/expressTaller/server.js>

¿Cuál fue el principal problema para resolver este ejercicio? ¿Cuál fue el problema que tuvo una de las personas en tu equipo? Explica cómo resolvió esa persona ese problema.

- **Ejercicio 1: Simplicidad**

¿ Recuerdas cómo **instalar un módulo** en node.js? Escribe la instrucción que se usa: Para instalar un módulo en node.js se debe usar la instrucción 'npm install'

- **Ejercicio 2: Express es**

Investiga qué es express (en el contexto de desarrollo web en nodejs). Explica a un **niñ@** qué es express, cómo se usa, para qué sirve. Escribe ese texto aquí. Incluye también **las referencias** de tu investigación.

Express es un framework web de Node.js, y también es el más popular. Varios problemas de desarrollo web pueden solucionarse con Express ya que ayuda a

manejar solicitudes con distintos métodos HTTP. Básicamente simplifica el uso web de Node.js.

Explicación para niño: Hay algo llamado Node, que es básicamente una herramienta que les ayuda a los programadores a crear páginas web y darles funcionamiento. Asimismo, hay algo llamado Express, el cual es una herramienta que facilita el uso de la otra herramienta Node. Express hace más simples las acciones que se deben hacer con Node, y con eso puede resolver problemas relacionados con Node.

https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction

Dile a una persona de la clase (que no esté en tu mesa) que te explique qué es express y qué ventajas te podría dar su uso. Graba el audio de la explicación.

¿Crees que realizó una buena explicación? ¿Por qué?

Sí, me parece que realizó una buena explicación ya que mencionó todos los puntos importantes de express con claridad.

- **Ejercicio 3: Desde un libro**

Investiga una definición en un libro (digital o físico) **de una API**. Incluye la referencia del libro.

En el contexto de software, un API (Application Programming Interface) es una herramienta en forma de interfaz que permite la interacción entre software de distintos tipos.

“API Architecture” por Matthias Biehl.

<https://books.google.com.mx/books?id=6D64DwAAQBAJ&lpg=PA15&ots=zb6l3tOa5x&dq=API&lr&pg=PA15#v=onepage&q=API&f=false>

Escribe un ejemplo de un endpoint de tu proyecto.

Un endpoint del proyecto podría ser la sección de búsqueda en la página web, que tendría la URL: /buscar/:keyword

Aquí se pueden buscar escuelas y donantes en la sección de búsqueda para la vista de administrador, o únicamente escuelas para la vista de donante.

Se debe usar el método Get para obtener la información de las escuelas y donantes a partir de un parámetro de búsqueda y mostrar bloques de información en específico. Se regresan los datos generales de la escuela o donante que contengan coincidencias con el parámetro de búsqueda.

Se despliegan los bloques con información general para escuelas o donantes.

- **Ejercicio 4: Instala express**

Escribe la instrucción que se usa para **instalar el módulo de express**:
npm install express

- **Ejercicio 5: Uso de express**

Ejecuta y explica que hace el siguiente código:

```
import express from 'express';

const app = express();

app.listen(1984, () => {
  console.log('Up and up');
});
```

Este código usa Express, crea una aplicación Express que se usa para configurar ajustes del servidor, e inicializa el servidor en el puerto 1984, y el mensaje de la consola corre cuando el servidor fue iniciado de manera exitosa.

- **Ejercicio 6: Uso de express++**

Ejecuta y explica que hace el siguiente código:

```
app.get('/bienvenida', (req, res) => {
  res.send('Esto no es una página html');
});

app.get('/otraBienvenida', (req, res) => {
  res.sendFile('bienvenida.html');
});
```

En caso de que alguno de estos te marque error, soluciona el problema y explica la forma de solucionarlo:

Este código realiza acciones para URLs específicas del servidor. La URL /bienvenida manda el texto 'Esto no es una página html', mientras que la URL /otraBienvenida debería enviar lo que esté dentro del archivo bienvenida.html. Sin embargo, por sí solo no funcionaría el código como está para la última URL. Para empezar, debe

existir el archivo bienvenida.html, preferentemente en la misma ruta donde está el archivo con el código actual. Además, una manera de mandarlo correctamente:

```
app.get('/otraBienvenida', (req, res) => {  
  res.sendFile('bienvenida.html', { root: process.cwd() });  
});
```

Lo que hace este código es especificar la ruta de la cual debe sacar el archivo, y en este caso este código sirve para cuando el archivo está en el mismo directorio que se está trabajando.

- **Ejercicio 7: Recuerdo de imagen**

Recuerdas que la imagen no se podía ver en <https://github.com/sgiomatec/act2025>. Un servidor entrega respuestas a solicitudes realizadas por un cliente. Cuando el cliente solicita una imagen (u otro tipo de archivo) el servidor tiene que saber responder.

Express nos propone crear un directorio para los archivos que queremos usar, por ejemplo imágenes o archivos de hojas de estilo.

Investiga qué es express.static. Escribe tu investigación.

Escribe tu investigación

Express.static es una función de middleware incorporada en Express, con el formato:

```
express.static(root, [options])
```

Sirve para usar archivos estáticos (que no cambian cuando la aplicación está corriendo) como imágenes y archivos CSS o de JavaScript. El argumento 'root' es donde se especifica el directorio de donde se quieren tomar los archivos. En caso de querer usar varios directorios, se debe usar la función múltiples veces.

- **Ejercicio 8: Película**

Pregunta a una persona de otra mesa su película favorita y sus razones.

¿Ya viste esa película?. ¿Te gusta? ¿La verías si no la has visto? ¿Por qué?

John Wick 4 porque es una película entretenida. La he visto y también me gusta.

- **Ejercicio 9: Transformación**

Usa express para transformar todo el código de servidor.js, nombra el archivo ahora servidor_express.js

Escribe aquí abajo el enlace al archivo en tu repositorio:

<https://seb-hdez-g.github.io/expressTaller/servidorExpress.js>

- **Ejercicio 10: Verbos**

En el contexto de desarrollo web, explica los siguientes verbos GET, POST, PUT, DELETE

GET se usa para pedir/tomar datos de un servidor, POST se usa para mandar datos al servidor para crear un nuevo recurso, PUT se utiliza para reemplazar o actualizar un recurso del servidor y DELETE se utiliza para borrar un recurso.

Explica los siguientes códigos, en el contexto de desarrollo web:

Estos son códigos de estatus y representan el resultado de un request HTTP.

1xx Este código es para estatus informacionales

2xx Este código es para estatus de éxito

3xx Este código es para estatus de redirección

4xx Este código es para estatus de error de cliente

5xx Este código es para estatus de error de servidor

Explica con un ejemplo (relacionado con tu proyecto/reto) cómo se atendería una solicitud de cada uno de esos verbos usando express. Incluye al menos una respuesta 404, 201 y 200.

En nuestra página web, un ejemplo de un código 1xx podría aparecer al momento de que un usuario publique archivos de verificación de identidad. Mientras el servidor tome el tiempo de procesar el request, se puede mostrar un estatus 102 `Processing` antes de mandar un código de éxito 2xx 201 `Created`, o un código de error de cliente 4xx 400 `Bad Request`.

Al momento de ingresar a la página web, cuando el usuario ingrese sus datos pre-registrados, podría recibir los códigos de estatus 2xx 200 OK si se validaron

correctamente los datos, o un status 4xx 401 `Unauthorized` si algo está mal escrito o falta llenar algún campo de información.

Un código de estatus 3xx 303 `See Other` se podría usar después de subir un archivo, para redireccionar al usuario a una página de éxito y evitar que por ejemplo se suban archivos múltiples veces por error.

Un código de estatus 4xx 404 `Not Found` se podría utilizar si por alguna razón la URL es modificada (accidentalmente se borra una letra o se agrega otra), lo cual causaría que la URL fuera inválida. En este caso al usuario se le mostraría el código de error para mostrar que la URL ingresada no existe.

Finalmente un código de estatus 5xx serviría para indicar que hubo un error de parte del servidor y que no depende del usuario, por ejemplo un estatus 500 `Internal Server Error`.

- **Ejercicio 11: Postman**

En equipo con tu mesa, instalen postman en cada una de sus computadoras. Tomen una misma foto en la que aparezcan todas las pantallas con postman abierto.

Foto aquí: (no pudimos tomarla todo el equipo)

Explica para qué sirve Postman

Postman es una plataforma API que contiene herramientas para diseñar, construir, probar, documentar y colaborar en APIs.

Prueba lo que hiciste en el ejercicio 10 con Postman. Pon una captura de pantalla usando el DELETE.

- **Ejercicio 12: Parámetros**

Inventa un ejercicio (que esté relacionado con escuelas/donantes) para que uses parámetros en las rutas (por ejemplo `/getEscuelas`)

Soluciona ese ejercicio

Escribe aquí abajo el enlace al archivo en tu repositorio:

<https://seb-hdez-g.github.io/expressTaller/express.js>

Compara ese ejercicio con el de otra persona de la clase. ¿Qué ejercicio ayuda mejor a comprender el uso de parámetros y por qué?

- **Ejercicio 13: npx**

Investiga con alguien más lo siguiente: `npx express-generator`. Importante haz un directorio diferente si quieres ejecutar el comando.

Escribe el nombre de ese alguien más: Pau

Explica lo que genera y hace ese comando.

Explica de manera básica y conceptualmente qué archivos se generaron.

Este comando sirve para crear de manera rápida un esqueleto de una aplicación. El comando crea un folder en el directorio en el que se encuentre. Se crean múltiples archivos de que funcionan para la aplicación, como archivos javascript, de estilo css, imágenes, etc.

- **Ejercicio 14: Explicar**

Explica a una niña o un niño de 10 años para qué sirven frameworks como Angular, React, Vue, o Svelte.

Los frameworks sirven para facilitar el desarrollo de una aplicación o una página web, ya que están hechos con herramientas que previamente fueron utilizadas y que pueden volver a ser aplicadas para otros proyectos.